LA-UR-21-28498
Approved for public release; distribution is unlimited.

Title: Random Forest vs. Mahalanobis Ensemble and Multi-Objective LDA

Author(s): Green, Andre Walter

Intended for: Progress report to sponsor

Issued: 2021-08-25

Random Forest vs. Mahalanobis Ensemble and Multi-Objective LDA

Presented 8/25/2021

Andre Green

Time & Space Complexity for Mahalanobis Ensemble & Random Forest

Mahalanobis Ensemble

For C classes and F-dimensional feature vectors:

Time complexity: O(C * (F^3))	C [FxF] matrix multiplications.
Space complexity: O(C * (F^2))	C [FxF] matrices.

Random Forest

For T trees with maximum depth D:

Time complexity: O(T * D)	T traversals of D-deep trees.
Space complexity: O(T * 2^D)	T D-deep trees.

If the random forest is checking multiple variables (say m) at each node of its trees, then the time & space complexities just change linearly: O(m * T * D) for time, O(m * T * 2^D) for space.

Random forests have the lower time complexity; Mahalanobis ensembles have the lower space complexity.

Random Forest vs. Mahalanobis Ensemble

An Isolation Forest was used to remove outliers prior to training & tests. As per discussion previously, the random forest was restricted to 30 trees, and a maximum depth of 8 was selected. [This puts each tree at 1KB minimum storage space]

RF Mean: **70-98%**
ME Mean: **83-99%**

Fold (1/9) | ME: 1.0 | RF: 1.0
Fold (2/9) | ME: 1.0 | RF: 0.982
Fold (3/9) | ME: 1.0 | RF: 0.982
Fold (4/9) | ME: 1.0 | RF: 0.982
Fold (5/9) | ME: 1.0 | RF: 0.982
Fold (6/9) | ME: 0.98 | RF: 0.982
Fold (7/9) | ME: 1.0 | RF: 1.0
Fold (8/9) | ME: 1.0 | RF: 0.982
Fold (9/9) | ME: 1.0 | RF: 1.0

[Name] : [mean, median, max, min] (9-fold SKF)

ME: **0.998**	1.0	1.0	0.982
RF: 0.988	0.982	1.0	0.982

RF : Max. Depth: 8 | Num. Trees 30
RF : Space Required: ~30.0 KB
ME : Space Required: ~1.98 KB

Dataset(s):
philadelphia_9_10_19
philadelphia_9_11_19_Act_1
philadelphia_9_11_19_Act_2
philadelphia_9_11_19_Act_5
philadelphia_9_11_19_Act_6

Fold (1/9) | ME: 0.873 | RF: 0.795
Fold (2/9) | ME: 0.849 | RF: 0.834
Fold (3/9) | ME: 0.878 | RF: 0.766
Fold (4/9) | ME: 0.863 | RF: 0.829
Fold (5/9) | ME: 0.844 | RF: 0.839
Fold (6/9) | ME: 0.853 | RF: 0.828
Fold (7/9) | ME: 0.863 | RF: 0.843
Fold (8/9) | ME: 0.868 | RF: 0.833
Fold (9/9) | ME: 0.853 | RF: 0.799

[Name] : [mean, median, max, min] (9-fold SKF)

ME: **0.86**	0.863	0.878	0.844
RF: 0.819	0.829	0.843	0.766

RF : Max. Depth: 8 | Num. Trees 30
RF : Space Required: ~30.0 KB
ME : Space Required: ~13.86 KB

Dataset(s):
ali

Fold (1/9) | ME: 0.815 | RF: 0.706
Fold (2/9) | ME: 0.829 | RF: 0.703
Fold (3/9) | ME: 0.818 | RF: 0.706
Fold (4/9) | ME: 0.836 | RF: 0.689
Fold (5/9) | ME: 0.829 | RF: 0.699
Fold (6/9) | ME: 0.836 | RF: 0.692
Fold (7/9) | ME: 0.808 | RF: 0.731
Fold (8/9) | ME: 0.829 | RF: 0.734
Fold (9/9) | ME: 0.864 | RF: 0.72

[Name] : [mean, median, max, min] (9-fold SKF)

ME: **0.829**	0.829	0.864	0.808
RF: 0.709	0.706	0.734	0.689

RF : Max. Depth: 8 | Num. Trees 30
RF : Space Required: ~30.0 KB
ME : Space Required: ~13.86 KB

Dataset(s):
25K_Cycles
51.4K_Cycles
101K_Cycles

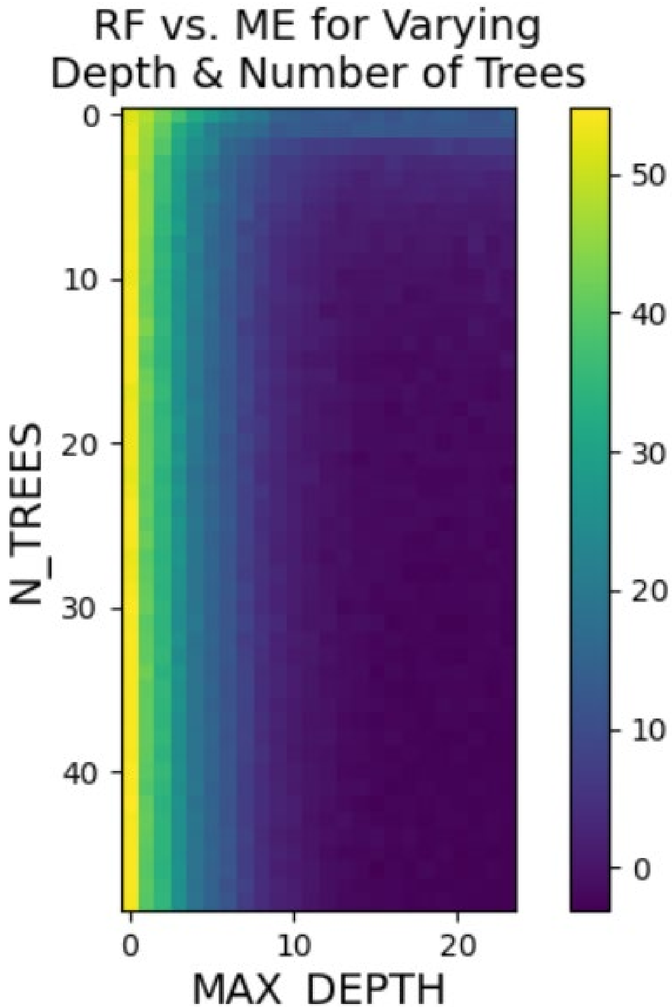
9-fold stratified cross-validation was used for testing.

Ali's 13 features (copied below) were used for classification: I will plan to try other features too.

Synthetic minority oversampling technique has not yet been applied here: I am not sure if it is appropriate for the Philadelphia dataset.

```
['Var_of_Accel_1', 'Var_of_Accel_2', 'Var_of_Accel_3',
 'Mean_of_PG_1', 'Mean_of_PG_2', 'Mean_of_PG_3',
 'Var_of_PG_1', 'Var_of_PG_2', 'Var_of_PG_3',
 'Slope_of_Angle', 'Pressure_Diff_Sum', 'Diff_Temp_Var', 'Pressure_Max']
```

Random Forest vs. Mahalanobis Ensemble



Using up to 50 trees with a maximum depth of 25 the results from the random forest are slightly better (2-3%: ME is approximately 90%, RF is 92-93% for sufficiently high number of trees and depths).

If there's either an efficient way to store these trees or they turn out to not be fully populated, it may be more performant to use random forests.

[Example]

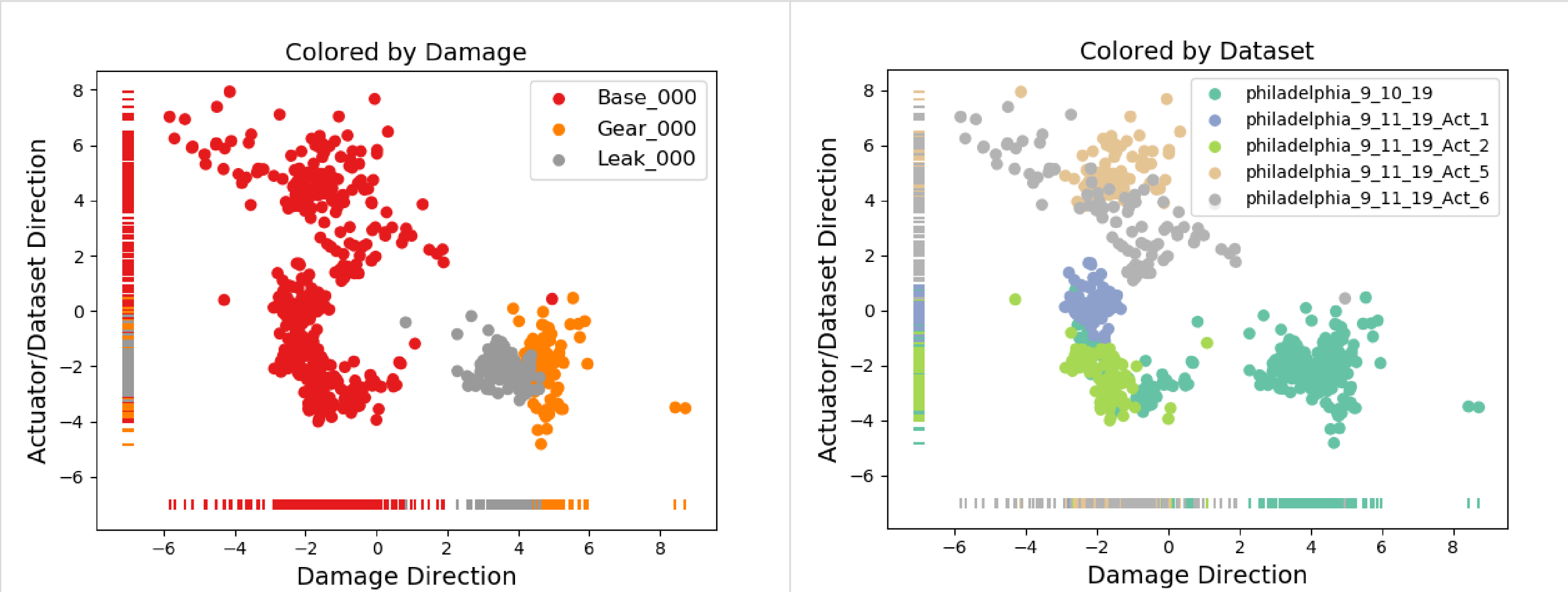
Fold (1/9) | ME: 0.902 | RF: 0.933
Fold (2/9) | ME: 0.901 | RF: 0.933
Fold (3/9) | ME: 0.884 | RF: 0.926
Fold (4/9) | ME: 0.919 | RF: 0.926
Fold (5/9) | ME: 0.873 | RF: 0.923
Fold (6/9) | ME: 0.887 | RF: 0.912
Fold (7/9) | ME: 0.898 | RF: 0.926
Fold (8/9) | ME: 0.908 | RF: 0.912
Fold (9/9) | ME: 0.891 | RF: 0.923

[Name] : [mean, median, max, min] (9-fold SKF)

ME: 0.896 0.898 0.919 0.873
RF: 0.924 0.926 0.933 0.912

RF : Max. Depth: 24 | Num. Trees 49
RF : Space Required: ~3211264.0 KB

Damage Type vs. Actuator [Supervised Dimension-Reduction]



Damage LDA Vector [1.0, 0.03, 0.00, 0.07, 0.08, 0.00, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.02]
Actuator LDA Vector [1.0, 0.44, 0.05, 0.01, 0.00, 0.04, 0.04, 0.04, 0.04, 0.04, 0.04, 0.04, 0.05]

dot: 0.92 [If 1, the two would be equivalent & perfectly correlated] (Normalized)
angle: 22.86 [If 0, the two would be equivalent & perfectly correlated] (In degrees)

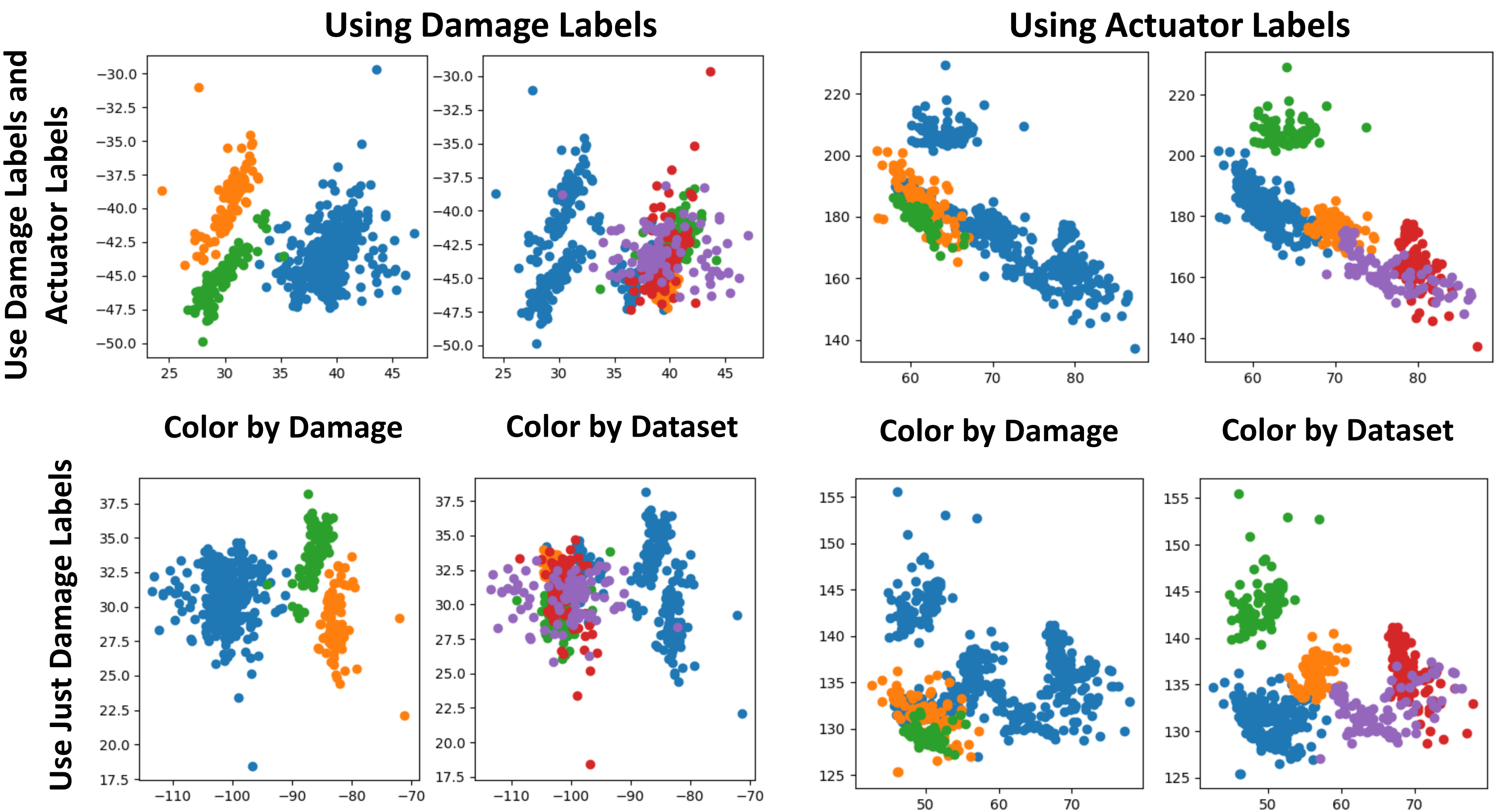
Variance of accelerator 2 is more important for separating actuators than the present damage types (Base/Gear/Leak). The pressure gauge means are more important for damage than actuator types.

```
['Var_of_Accel_1', 'Var_of_Accel_2', 'Var_of_Accel_3',
 'Mean_of_PG_1', 'Mean_of_PG_2', 'Mean_of_PG_3',
 'Var_of_PG_1', 'Var_of_PG_2', 'Var_of_PG_3',
 'Slope_of_Angle', 'Pressure_Diff_Sum', 'Diff_Temp_Var', 'Pressure_Max']
```

C:\Andre_Green_FY2020\during_corona\TW6100_Actuators\actuator_vs_damage.py

Dual-Objective Linear Discriminant Analysis

Normal LDA solves **eig(between * inv(within))**, whereas dual-objective linear discriminant analysis solves **eig(Between_A * inv(Within_A) * Within_B * inv(Between B))**.



https://en.wikipedia.org/wiki/Data_anonymization